Improving Availability and Confidentiality of Shared Data under the Multi-cloud Environment

Kaiying Feng, Junxing Zhang
College of Computer Science
Inner Mongolia University
Hohhot, China
e-mail: junxing@imu.edu.cn

Abstract—**Current providers of the cloud storage service often ensure the data confidentiality by encrypting the file content and guarantee the data integrity by verifying the hash value of the file. However, when the cloud storage service fails, the availability of the user data cannot be guaranteed and nor can the cloud sharing function of the user data be supported. In addition, users have to give the provider of the cloud storage service full trust in existing schemes. Once the provider system is hacked or becomes untrustworthy, the confidentiality of the user data will be threatened. In order to solve these problems, this paper proposes a scheme for securely storing and sharing data based on the proxy re-encryption algorithm in the multi-cloud environment. In this scheme, the multi-cloud storage is designed to prevent the failure of any single cloud, the symmetric encryption algorithm is used to encrypt user files, the encryption key is reliably distributed with the Shamir's threshold secret sharing scheme, and finally the proxy re-encryption algorithm is leveraged to support sharing of the encrypted data via clouds. The prototype of the scheme is implemented in the Java development environment and is evaluated under the simulated multi-cloud environment. Our experimental results show that, the time cost of the Shamir's secret partitioning process and symmetric encryption process almost can be negligible when the key size is as long as 386 bytes, and the proxy re-encryption process takes about 1.6 seconds in average.**

Keywords-data partition; encrypted data sharing; proxy re-encryption

I. INTRODUCTION

With the cloud storage service, users can access, manage, and share their data conveniently. In fact nowadays people are accessing the cloud data with networked devices at any time from any place, without having to care about the details of the cloud platform. However, the current cloud service for storing and sharing user data has the following problems:

A. Availability

Cloud service providers (CPSs) may suspend services unexpectedly. For example, in 2015 many downtime incidents of CSPs, including Microsoft Azure, Amazon Web Services, and Google Compute Engine, are reported.

B. Security

As the cloud needs to store huge amounts of data, the user data is likely to be stored in plain text from the performance considerations. Then the hackers and even the CSP can peep at the users' data. Even if the CSP has encrypted the cloud data, CSP also has the ability to acquire contents of plain text.

C. Encrypted Data Sharing

When the user needs to share the existing encrypted data in the cloud, it is necessary to disclose the key used for encryption in order to let the other side obtain the plaintext, which will also lead to data leakage.

In order to solve these problems, many researchers have put forward some new research on encryption algorithm. At the 1998 European Conference on Cryptography, Blaze et al. [1] first proposed the proxy re-encryption scheme. In the proxy re-encryption, the proxy turns the authorizer's ciphertext encrypted with a public key into recipient's ciphertext encrypted with another public key by the switching key of the authorizer. After this, various improved versions of proxy re-encryption have come to being [2-5]. In 2015, David et al. [6] improved proxy re-encryption scheme based on NTRU encryption algorithm. NTRU algorithm has the ability to resist quantum computing attacks, and the encryption algorithm is simple, fast in calculation and small in storage space, which is more suitable for data encryption of resource-limited client. In the same year, Hua Deng et al. in [7] ingeniously improved the proxy re-encryption algorithm to select different encryption algorithms for different computing power devices. These studies had improved the security of data to a certain extent. However, they are still aimed at single cloud storage. When the cloud service providers cannot provide services, users still need to bear the loss of data unavailable.

In order to solve this problem of single cloud, Jae et al. in [8] integrated the mainstream of cloud services and provided a unified interface for user by integrating heterogeneous cloud services. Singh et al. in [9] proposed a multi-cloud storage model SCMCS to provide data partition function for users. HU et al. in [10] used the data redundancy to repair the cloud data by other available cloud storage providers.

Based on related research, in view of the cloud storage and sharing of data, this paper designs a scheme under multi-cloud environment. The scheme encrypts the user data with a symmetric encryption algorithm and then uploads the data to multiple cloud storage servers. The Shamir's secret sharing scheme is used to divide the encryption key, and then the key fragments are processed respectively by the proxy re-encryption algorithm and uploaded to the corresponding

978-1-5090-4499-3/17/$31.00 ©2017 IEEE

cloud. The scheme will use enough key fragments to recover the initial key and decrypt the file when the user needs to obtain the cloud data. When the user needs to share the encrypted files, the scheme will convert the ciphertext based on the proxy re-encryption algorithm, so that it can be shared in the form of decrypted ciphertext of the sharer.

The contributions of this paper are as follows: our scheme increases the redundancy of data based on the secret sharing scheme to overcome the invalidation of single cloud service, which can ensure the availability of cloud data. By using the proxy re-encryption algorithm, our scheme can transform the encrypted data in the cloud, and support the cloud data sharing function without affecting the security of data. In addition, a single cloud stores the ciphertext and key fragments because the data encryption occurs on the client side, and such design may avoid being peeped as well. Finally, this paper achieves the prototype of this scheme based on java development environment, and evaluates the scheme under the multi-cloud environment simulated by the local disk. The experimental results show that the time cost of symmetric encryption and key partitions based on Shamir's secret sharing is at millisecond level, which can be negligible. In the most time-consuming proxy re-encryption process, the parameter initialization process takes close to 6s, but this step occurs only once, and the rest processes all take about 1.6s except the initialization step.

The rest of this paper is organized as follows. The following section provides the background of the Shamir's Secret Sharing Scheme and the Proxy Re-Encryption Scheme. Section III presents the design details of our system. This paper evaluates the proposed system in Section IV. Finally, Section V concludes the work.

II. BACKGROUND

In order to give readers a better understanding of the design of this paper, this section provides a brief introduction of the Shamir's Secret Sharing Scheme and the Proxy Re-Encryption Scheme.

A. *Shamir Secret Sharing Scheme*

In recent years, scatter storage of key has become a trend of key management, which helps to solve the loss of user data which is caused by loss of user's key, and it is important to the security of computer and network in theory and practice. The (t, n) threshold key sharing scheme was proposed by A. Shamir and G. R. Blakley in 1979. Among them, the polynomial-sharing scheme proposed by Shamir is easy to understand and implement, and it includes three stages: 1) initialization, 2) secret distribution, and 3) secret recovery.

Shamir's scheme satisfies the security requirements of the secret sharing, that is: 1) no less than t shadow secrets can recover the master secret; 2) less than t shadow secrets cannot get any information of the master secret. In this scheme, the key segmentation is performed based on the secret sharing scheme of Shamir. The specific process is shown in figure 1. The original information is divided into n parts, and the original data can be reconstructed only if there are t or more sub-information.

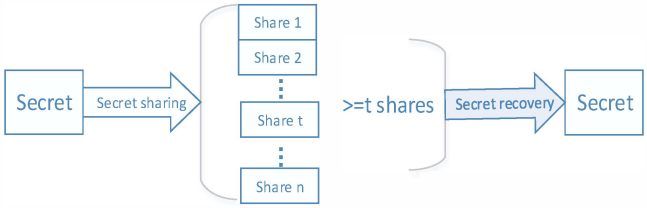

Figure 1. The flow chart of Shamir's secret sharing

B. *Proxy Re-Encryption Scheme*

Proxy re-encryption is a key conversion scheme between the ciphertexts. In recent years, cloud computing and cloud storage are becoming more and more popular, and they has been closely linked with our daily life. Sometimes users' data need to be transmitted and shared in cloud. The proxy re-encryption is a good ciphertext conversion scheme, which is suitable for the data sharing of cloud. The scheme of this paper adopts this ciphertext conversion scheme. Figure 2 depicts the workflow of proxy re-encryption in the cloud.

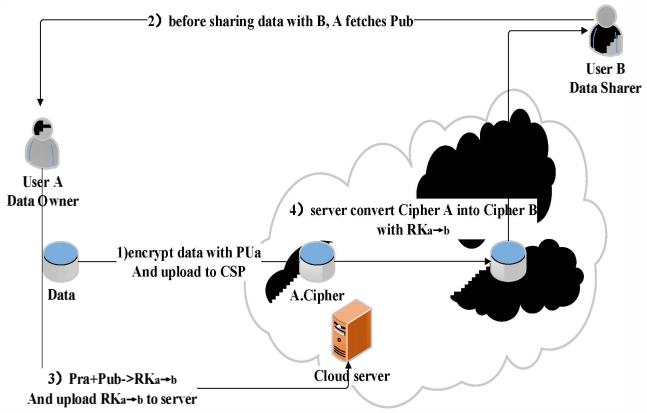

Figure 2. Proxy re-encryption diagram

In cloud computing, the cloud computing service provider is as a proxy, and user A cannot fully trust the cloud computing service provider, so he or she needs to encrypt the data in their local machine with a public key P_a and then upload the data to the cloud storage. In this way, the cloud computing service providers cannot be able to get cleartext information of the data. The data only can be decrypted with the private key S_a of user A. When the user A needs to share the data with the user B, he or she can calculate a conversion key R_k according to some of their own information (such as the private key Pa) and the public key P_b of user B. The cloud computing service provider uses the conversion key R_k to re-encrypt the ciphertext of user A to obtain the ciphertext for user B, then user B can easily download the ciphertext data from the cloud and decrypt it using his or her own private key Sb. Thus the entire lifecycle of the data in the cloud is stored in ciphertext and the cloud service provider cannot know the private keys of user A and user B, so the

cloud computing service provider cannot obtain plaintext of the data. Therefore, the security of user's data shared in the cloud is ensured well.

III. Overall Design and Workflow

This paper focuses on the encryption uploading of user data and cloud data sharing two modules, so how to design compatible client for cloud storage service interface given by different CSP will not be discussed in this paper. Here, this paper designs a scheme structure of data storage based on proxy re-encryption under multi-cloud environment. The scheme is divided into the following sections:

a) Symmetric encryption of user data

b) Symmetric encryption key segmentation based on SSSS

c) The design of the encryption process for the segmentation key

d) Cloudy storage of ciphertext

e) PRES-based data sharing design

f) Cloud data ciphertext downloading and clear text recovery

Next, this paper will detail the specific workflow of this scheme.

As shown in the Figure 3, the client-server model is adopted in this scheme. The client includes DO (Data Owner), KDC (Key Distribution Center) and DS (data sharer). Server-side includes PS (Proxy Server), MCSS (Multi-Cloud Storage Server) and SMS (System Management Server).

The data owner can share the data he owns with his authorized users, and the data sharer also can request the data owner to authorize and obtain the data. In order to avoid the risk of the private key transmission in the network as much as possible, the scheme puts the KDC on the client. Proxy server is responsible to transfer the encrypted data of data owner into the ciphertext form which can be decrypted by the data sharer. The cloud storage server is responsible for storing the user's data and ensuring the robustness and security of the stored data. System management server mainly stores some public information of the system, such as the user's public key information and system public parameters for users' access.

In this paper, the data storage based on proxy re-encryption under multi-cloud environment includes nine steps: PRE initialization, key generation, key segmentation, data encryption, data uploading, transform key generation, PRE re-encryption, data downloading and data recovery.

a) PRE initialization: A system security parameter is selected to be as the input to generate the corresponding public parameter. The public parameter is stored in the system management server for user access and it is also used as parameters for the user key generation algorithm and data operation, namely the PRE initialization process.

b) Key generation: 1) The symmetric encryption algorithm key DES.key is randomly generated by the data owner based on DES, and used for data encryption and decryption. 2) The public-private key pair (pk1, sk1) and (pk2, sk2) of both the data owner and the data sharer need to be calculated by the key distribution center (KDC) according to the system parameters and user identity information, and stored in the client. At the same time, the public key pk1 and pk2 of their own users are uploaded to the system management server for other users' access.

c) Key segmentation: DES.key is divided into n sub-DES.keys by Shamir secret sharing scheme.

d) Data encryption: The symmetric encryption is more efficient than asymmetric encryption but less secure.

Because the amount of user data is often large, the efficiency would be too low if using asymmetric encryption. Therefore, symmetric encryption is used to encrypt user data while the symmetric encryption key is encrypted by using asymmetric encryption, and integrating the design of re-encryption algorithm, the final specific encryption process is as follows: 1) DES encryption. The data owner encrypts his data through DES.key as ciphertext. 2) PRE first-level encryption. The n sub-DES.keys are encrypted to generate n DES.key.secret through the public key pk1 of the data owner.

e) Data upload: 1) In order to prevent the failure of single cloud, our scheme uploads the multiple copies of the ciphertext to multiple clouds for storage. 2) In the same way, our scheme uploads the n sub-DES.key.secret to multiple clouds for storage.

f) Transform key generation: When a data sharer wants to obtain an encrypted data in a cloud storage server, he first needs to send a request to the data owner for data access and the data owner first verifies the legitimacy of the data sharer after he or she receives the request. If the data sharer is legitimate, the data owner needs to visit the system management server to obtain the public key pk2 of the data sharer and then combining with owner's private key sk_1, the transform key r_k is calculated by using the key generation algorithm of proxy re-encryption.

g) PRE re-encryption: The data owner sends the generated transform key r_k to the proxy server. The proxy server re-encrypts the n sub-DES.key.secret in accordance with the re-encryption algorithm, and sends the n sub-DES.key.re-secret and the corresponding data ciphertext to the cloud storage server directory of the data sharer. In this way, the data sharer becomes the data owner of the ciphertext.

h) Data download: The data owner visits the cloud storage server and downloads any of the k pieces of sub-DES.key.re-secret and any a ciphertext.

i) Data recovery: 1) PRE first-level decryption. The data owner decrypts the k DES.key.re-secret into k DES.key by using his or her own private key, and then uses the Shamir secret sharing algorithm to reconstruct the original symmetric key DES.key. 2) DES decryption. Data sharer uses DES.key to decrypt the ciphertext into plaintext of the data.

At this point, the basic process of this scheme in this paper is introduced. The data sharer of this scheme is not only one person, and all the users who apply for authorization of the data owner can safely obtain the confidential information of the data owner. The authorization process is also secure, so this scheme has good scalability and security for data sharing.

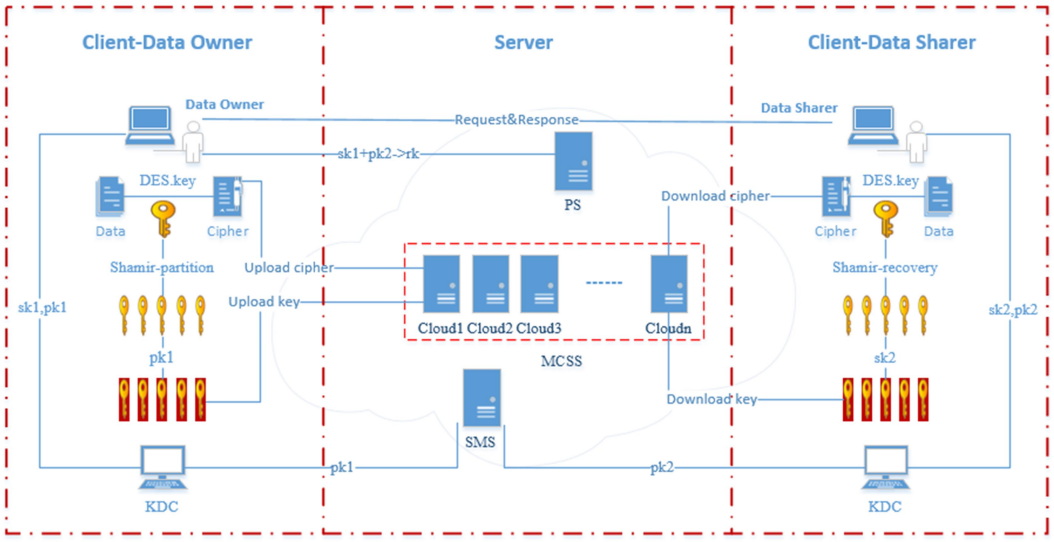

Figure 3. Proxy re-encryption diagram

IV. Implementation and Performance Analysis

This paper has achieved the scheme prototype based on java development environment. Because this design is more focused on the study of user data's encryption upload and cloud data sharing two modules, the paper does not design compatible client for cloud storage service interface given by different CSP, but adopts the cloud storage server simulated by the local disk, ignoring the details of file upload.

The scheme will be evaluated from the following two aspects: security analysis and time cost.

A. Safety Analysis

The main process of the scheme includes nine steps: PRE initialization, key generation, key segmentation, data encryption, data upload, transform key generation, PRE re-encryption, data download, and data recovery.

Among them, key generation, key segmentation, data encryption and data recovery occur in the client, and the client's security is easy to strengthen.

The data uploading, transform key generation and data downloading may be subject to eavesdropping and brute force because these processes need to transmit data through the network. However, the data has been partied and encrypted in this scheme, thus the attacker cannot get valid information. In addition, although there may be risks of man-in-the-middle attacks, hackers may illegally tamper with transmitted data or act as impostors, the CA scheme adopted by this scheme can effectively prevent this kind of risks.

PRE initialization and PRE re-encryption these two processes occur on the server side. PRE initialization is to generate the corresponding public parameters for user access, so it does not require encryption because there is no security risk. However, the PRE re-encryption process is visible to the cloud service provider, but the proxy re-encryption is used, which can avoid the storage and transmission of user's private key. In addition, the user's data in the cloud is stored in ciphertext, and the key DES.key is a distributed storage in the cloud after Shamir segmentation with asymmetric encryption. Therefore, even if the cloud service provider obtains user's private key of asymmetric encryption, it is also difficult to get DES.key, thus it is almost impossible to acquire the original data. However, in practice, our scheme needs to understand that the user's private key of asymmetric encryption exists only in the client, so the cloud service provider cannot get it.

Above all, this paper has analyzed the security of the nine processes of the scheme, and each process has a high security, so the scheme is safe and reliable.

B. Time Cost Analysis

In this paper, 100 sets of data with different contents are selected for experiment. The data size of each set is 1MB. In order to carry out stress test, the key length of this scheme is 386bytes.

As DES encryption and decryption technology is relatively mature, the encryption and decryption rate is up to 1MB/s, and time cost of SSSS scheme is also at the ms level, then it can be ignored, so this paper mainly evaluates the time cost from the following parts: (1) PRE initialization (2) PRE first-level encryption (3) PRE re-encryption (4) PRE first-level decryption (5) transform key generation.

This paper examines the time cost of the above five phases respectively, and the specific time is shown in Table 1:

TABLE I. The Time Cost of Different Phases

Phase	Time average (ms)	standard deviation
Initialization of PRE	6345.7	4099.6
PRE first-level encryption	1610.8	68.4
PRE re-encryption	104	18.4
PRE first-level decryption	1581	40.4
Transform key generation	1509.6	36.7

From the experimental results in Table 1, it can be seen that the PRE initialization has a long time cost comparing

with other processes. However, in this scheme, the PRE initialization occurs only once, so it does not require repetitive operations, but the other operations are performed depending on the sharing frequency. Therefore, in order to improve the security of data storage and sharing in multi-cloud environment, the time cost of PRE initialization (approximately 6.3 s) is acceptable. In addition, it's observed that the standard deviation of PRE initialization is also large, which may because that the experimental data contents of 100 sets generated randomly is extremely disordered, resulting in initialization time cost is very unstable.

In order to better analyze the time cost distribution of each phase of the scheme in Table 1, this paper makes Figure 4, which omits the PRE initialization for it takes a relatively long and unstable time.

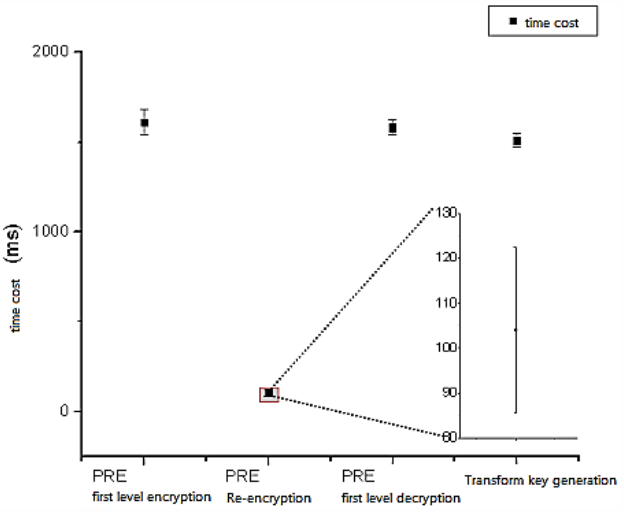

Figure 4. The box diagram of time cost in scheme phases

From observing Figure 4, our analysis shows that the time cost of PRE first-level encryption, PRE re-encryption, PRE first-level decryption or transform key generation is generally very short, which is less than 2s. In addition, PRE re-encryption phase takes about 0.1s.

From the above analysis, it can be seen that our scheme costs little time, and it has high efficiency, which has potential application.

V. CONCLUSION

In view of the availability and security problems of the current cloud storage, this paper proposes a secure storage and sharing scheme in multi-cloud environment with Shamir secret sharing scheme and proxy re-encryption scheme. The analysis shows that the scheme has high security, which can prevent common eavesdropping and brute force attacks. At the same time, the evaluation data shows that the time cost of the scheme is also within the acceptable range of users, so it has a certain degree of feasibility and application prospect. In the follow-up work, we will further optimize the scheme to make it easier to operate, and to achieve the connection and interaction with the existing commercial clouds, so that the scheme is more suitable for practical application.

ACKNOWLEDGMENT

This work was supported in part by the National Natural Science Foundation of China (Grant No.61261019), the Inner Mongolia Autonomous Region Natural Science Foundation (Grant No.113113), and the Program of Higher-level talents of Inner Mongolia University.

REFERENCES

[1] M. Blaze, G. Bleumer, and M. Strauss. Divertible protocols and atomic proxy cryptography. EUROCRYPT, 1998.

[2] Edna Milgo. A Secure Unidirectional Proxy Re-Encryption Using Identity and Secret Key Exchange. ACMSE '09

[3] G. Ateniese, K. Benson, and S. Hohenberger. Key-private proxy re-encryption. Topics in Cryptology - CT- RSA 2009.

[4] Y. Aono, X. Boyen, L. T. Phong, and L. Wang. Key-private proxy re-encryption under LWE. In Progress in Cryptology IN DOCRYPT 2013

[5] G. Ateniese, K. Fu, M. Green, and S. Hohenberger. Improved proxy re-encryption schemes with applications to secure distributed storage. In Proceedings of the 12th Annual Network and Distributed System Security Symposium, 2005.

[6] David Nuñez, Isaac Agudo, and Javier Lopez. NTRUReEncrypt: An Efficient Proxy Re-Encryption Scheme Based on NTRU. ASIA CCS'15

[7] Hua Deng, Qianhong Wu, Bo Qin, et al. Asymmetric Cross-cryptosystem Re-encryption Applicable to Efficient and Secure Mobile Access to Outsourced Data. ASIA CCS'15

[8] Jae Yoon Chung, Carlee Joe-Wong, Sangtae Ha, James Won-Ki Hong, and Mung Chiang. CYRUS: Towards Client-Defined Cloud Storage. EuroSys'15

[9] Singh Y, Kandah F, Zhang W. A secured cost-effective multi-cloud storage in cloud computing. Computer Communications Workshops (INFOCOM WKSHPS), 2011

[10] Hu Y, Chen H C H, Lee P P C, et al. NCCloud: Applying network coding for the storage repair in a cloud-of-cloud. USENIX FAST, 2012